OFFIS

# QP 3.1

## URANOS-X

Mathias Uslar
OFFIS - INSTITUT FÜR INFORMATIK

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN
fir an der RWTH Aachen
Fraunhofer IEM
OFFIS
URANOS X
GAIA-X für die Produktion

# Inhalt

1 Architektur und Referenzbegriff ..... 1
1.1 Grundlagen und Motivation ..... 1
1.2 Referenzarchitektur als Begriff ..... 3
1.3 Wiederverwendung – das Pattern oder auch Muster ..... 5
1.4 Der Referenzbegriff ..... 6
1.5 Dokumentationserfordernisse ..... 8
2 RAMI – Referenzmodell Industrie 4.0 ..... 11
2.1 Empfehlungen ..... 14
3 Literatur ..... 15

Abbildung 1: Begrifflichkeiten im Kontext Architektur 4
Abbildung 2: ISO IEEE/IEC 42010 Metamodell 10
Abbildung 3: RAMI 4.0 13
Abbildung 4: Mögliche Toolchain für RAMi 4.0 Engineering 14

# 1 Architektur und Referenzbegriff

## 1.1 Grundlagen und Motivation

Mit dem Wandel hin zu individueller industrieller Fertigung steigt die Anzahl der Anlagen und Fabriken in der Wertschöpfungskette und damit die Zahl der Teilnehmer. Zwischen den Teilnehmern finden in dieser neuen Wertschöpfungskette auch neue, bisher nicht da gewesene Interaktionen statt und es stehen neue Dienste zur Verfügung. Durch die neuen Möglichkeiten und Vernetzungen an sich können wiederum neue, heute noch nicht vorhersehbare Möglichkeiten entstehen.

Zur Verwirklichung dieser zukünftigen Produktion, zumeist Industrie 4.0 genannt, ist eine entsprechende, offene und flexible Informationsinfrastruktur nötig, mit der die steigende Anzahl und Geschwindigkeit der Interaktionen realisiert und die, damit verbundenen, umfangreichen Daten verarbeitet werden können.

Zum einen erfordert dabei beispielsweise der Umfang des Systems bzw. des System of Systems an sich und die große Anzahl der daran beteiligten Akteure eine systematische Vorgehensweise schon zum Entwurfszeit, die zu einer gesteuerten Evolution der einzelnen, vernetzten Produktionsstätten beiträgt.

Die Entwicklung solch komplexer Systeme ist dabei eine nicht-triviale Aufgabe und insbesondere eine große Herausforderung wie Industrie 4.0 lässt sich i. d. R. nur durch interdisziplinäre Zusammenarbeit mehrerer Expertendomänen realisieren.

Dafür existieren in Literatur und Praxis bereits etablierte Vorgehensmodelle, wie beispielsweise das klassische Wasserfallmodell, das V-Modell oder den „Rational Unified Process“ (RUP). Diese Vorgehensmodelle teilen den Gesamtentwicklungsprozess in mehrere Phasen ein, in denen anhand ausgewählter Methoden mit bestimmten, wiederkehrenden Aspekten, Lösungen für die Problemstellung entwickelt werden. Unter anderem bedingt durch ein unterschiedliches Verständnis und eine unterschiedliche Verwendung von Terminologie in verschiedenen Fachgebieten, ist eine Einordnung und Dokumentation des gemeinsamen Wissens sinnvoll. Eine Hilfestellung hierfür, die auch in vielen Vorgehensmodellen berücksichtigt wird, bieten Referenz(architektur-), welche eine Domäne und darin bestimmte und bereits gelöste, wiederkehrende Probleme abstrakt und wiederverwendbar beschreiben. Architekturen an sich, und Referenzarchitekturen im Besonderen, stellen u. a. auch die Basis für ein gemeinsames Verständnis, Verhandlungen und Kommunikation im Allgemeinen mit den an der Entwicklung Beteiligten dar, und legen frühzeitige Entwurfsentscheidungen für die Systementwicklung fest. Die in Referenzarchitekturen verwendeten Konzepte und Lösungen zeichnen sich insbesondere dadurch aus, dass sie sich bewährt haben und mit den Referenzarchitekturen zur Entwicklung konkreter Instanzlösungen beitragen sollen.

Durch den mit der Nutzung bewährter Konzepte einhergehenden fokussierten Wiederverwendungsaspekt, versprechen Referenzarchitekturen auch die Konstruktion mehrerer gleichartiger Lösungen, mit einem geringeren Zeitaufwand und Kosten im Vergleich zu einer Neuentwicklung. Zudem kann durch die Verwendung bewährter Lösungskonzepte die Qualität der Lösung bzw. der daraus abgeleiteten Architektur erhöht werden.

Mit einer Softwarearchitektur soll die Funktionalität des zu entwickelnden Systems umfänglich, ganzheitlich beschrieben und partitioniert werden. Die Dekomposition des Problembereiches hilft dabei, die Komplexität großer Systeme handhabbar zu machen, da nur einzelne Teilbereiche betrachtet werden müssen. Einzelne Teile zusammengehöriger Funktionalität (z. B. Komponenten)

realisieren dann im Zusammenspiel das Gesamtsystem. In URANOS-X betrifft dies im Besonderen den neuen Bereich der Data Spaces.

Anhand der Aufteilung der Software in unterschiedliche Teile im Rahmen der Architektur steht u. a. eine Planungsgrundlage für die Softwareentwicklung zur Verfügung, die speziell für die Zeitplanung, Arbeitszuordnung, Kostenanalyse sowie das Risikomanagement und zur Analyse der benötigten Fähigkeiten verwendet werden kann. Hierbei bilden insbesondere die üblichen Artefakte der Architektur (Module, Komponenten, Schnittstellen, Interaktionen etc.) und deren Abhängigkeiten eine Grundlage für die Planungen. Durch die Aufteilung des Problembereiches in verschiedene softwaretechnische Artefakte und deren Beziehungen untereinander sorgt der Prozess der Architekturentwicklung idealerweise für Integrität der resultierenden Architektur. Änderungen am System schlagen sich unter optimalen Umständen in Änderungen einzelner architekturrelevanter Artefakte mit hoher Kohäsion (z. B. einer oder mehrerer Komponenten) nieder und betreffen daher nicht das gesamte System.

Entsprechende Anpassungs- und Erweiterungspunkte (Variationspunkte für Variantenmanagement) können bei der Entwicklung der Architektur eingeführt werden, um spätere Änderungen und Erweiterungen an bestimmten Stellen möglichst einfach zu gestalten, bzw. um aus einer gemeinsamen Architektur mehrere Produktlinien entwickeln zu können.

Der Einbezug von mit der Wartung betrauten Stakeholdern in den Architekturentwicklungsprozess kann schließlich auch Wartungsaspekte im Vornherein berücksichtigen und diese somit unmittelbar in die Architektur einfließen, wodurch spätere Wartungskosten möglicherweise reduziert werden können. Angemessen spezifizierte Teile der Architektur können ferner Einheiten für Wiederverwendung darstellen, die in weiteren Kontexten (Architekturen bzw. Implementierungen) eingesetzt werden. Durch die Explizierung der architektonischen Elemente und eine getrennte Entwicklung ist auch eine Verwendung einzelner Elemente möglich, die wiederum extern entwickelt worden sein können (Commercial Off the Shelf (COTS) Software) und in die Lösung einfließen können.

Im Kontext des Forum SCI 4.0 wurde mit dem RAMI eine Architektur entwickelt, die als eine Referenz für das zukünftige Fabrikökosystem mit den entsprechenden Wirkungsbereichen dienen soll. Dabei wird insbesondere versucht Probleme der Domäne wie Altsysteme, proprietäre Protokolle oder gescheiterte Anwendungen anzugehen, sodass fortschrittlichere Anwendungen ermöglicht werden können. Das RAMI beinhaltet zudem eine empfohlene Vorgehensweise, mit der neue, sofort integrierbare Anwendungen ermöglicht werden sollen (plug-and-play) und mit der Alt-Schnittstellen durch aktuell eingesetzte ersetzt werden sollen. Mit standardisierten Komponenten, die von verschiedenen Herstellern im Wettbewerb angeboten werden, sollen Ersparnisse beim Auf- und Ausbau der Infrastruktur sowie bei der Wartung der Infrastruktur erzielt werden.

Ebenso können die Entwicklungskosten durch das Aufsetzen auf RAMI Komponenten und Verwaltungsschalen Komponenten reduziert werden. Die vorgegebenen Integrationsmöglichkeiten sollen so insgesamt Fehlinvestitionen, beispielsweise in Eigenlösungen, vermeiden. Darüber hinaus soll mit dem RAMI die Möglichkeit geschaffen werden zunächst im Kleinen mit dem Aufbau zu beginnen und diesen später zunehmend zu steigern. Eine Gesamtsystemarchitektur, wie sie durch die RAMI angestrebt wird, ist letztlich auch dazu nötig, um das künftige Produktionssystem ganzheitlich absichern zu können. Durch die dort angestrebten strukturierten Ansätze zum Systemmanagement soll zudem eine höhere Robustheit erreicht werden.

## 1.2 Referenzarchitektur als Begriff

Gerade im Kontext einer Entwicklung mit einer Vielzahl Beteiligter wie dem emerging Topic der Data Spaces sollte idealerweise vorab ein gemeinsames Verständnis über die grundlegenden Begriffe herrschen. Zum Begriff der Referenzarchitektur existiert in der Literatur keine allgemein verwendete, eindeutige Definition. Daher werden hier zunächst verschiedene Definitionen zum Begriff Architektur vorgestellt, der Begriff Referenz sowie Referenzarchitektur näher betrachtet, um ein grundlegendes Verständnis zu schaffen, und schließlich eine Arbeitsdefinition für den Begriff Referenzarchitektur entwickelt.

Für den Begriff Architektur existieren im Softwareumfeld mehrere Definitionen und Verständnisse, wie z. B. Architektur als Blaupause, Architektur als Literatur, Architektur als Sprache oder Architektur als Dokumentation von Entscheidungen. Im Kontext etwas allgemeineren Kontext Software-intensiver Systeme wird allerdings der IEEE Standard 1471-2000 häufig herangezogen. Als ein Software-intensives System werden dabei alle Systeme angesehen, in denen Software essenziell zum Entwurf, der Konstruktion, dem Deployment und der Evolution des Systems als Ganzen beiträgt. Nach IEEE ist Architektur (im englischen Original)

*„The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.“*

Dieser Definition folgend ist Architektur die Ausgestaltung der Komponenten mit ihren Beziehungen zueinander und zur Umwelt sowie der Prinzipien, die den Entwurf und die (Weiter-)Entwicklung bestimmen. Dieser Architekturbegriff ist bewusst wenig spezifisch gehalten und lässt sich über Software hinaus z. B. genauso auf Unternehmen (bzw. Unternehmensarchitektur) anwenden. Ein System aus Komponenten (z. B. logische oder physische Anwendungen, Systeme im traditionellen Sinne, Subsysteme, Systeme aus Systemen, ganze Unternehmen oder andere Aggregationen) befindet sich in einer vom System abgrenzbaren Umwelt. Durch Beziehungen der Komponenten untereinander sowie zur Umwelt kann eine Beeinflussung der Komponenten erfolgen.

Eine weitere, häufig referenzierte Definition stammt in diesem Zusammenhang von Bass. Darin wird weiter zwischen System- und Softwarearchitektur differenziert, da beim Entwurf einer Softwarearchitektur meist auch Hardwareaspekte berücksichtigt werden müssen, um z. B. kurze Antwortzeiten garantieren zu können.

Die Kombination aus Softwarearchitektur und der dafür benötigten Hardware wird daher als Systemarchitektur bezeichnet. Die eingeführte Trennung der beiden Architekturbegriffe soll speziell die mit dem Softwareentwurf verbundenen Freiheiten betonen, die im Falle von Hardware weniger ausgeprägt sind. Das heißt die Hardwarewahl ist beispielsweise häufig vorgegeben (z. B. beliebiger Rechner über das Internet), wird von anderen spezifiziert (z. B. aus ökonomischen Gründen oder aufgrund der Einhaltung von technischen und organisatorischen Standards) oder es ist wahrscheinlich, dass sich diese im Laufe der Zeit bedingt durch technischen Fortschritt ändern wird.

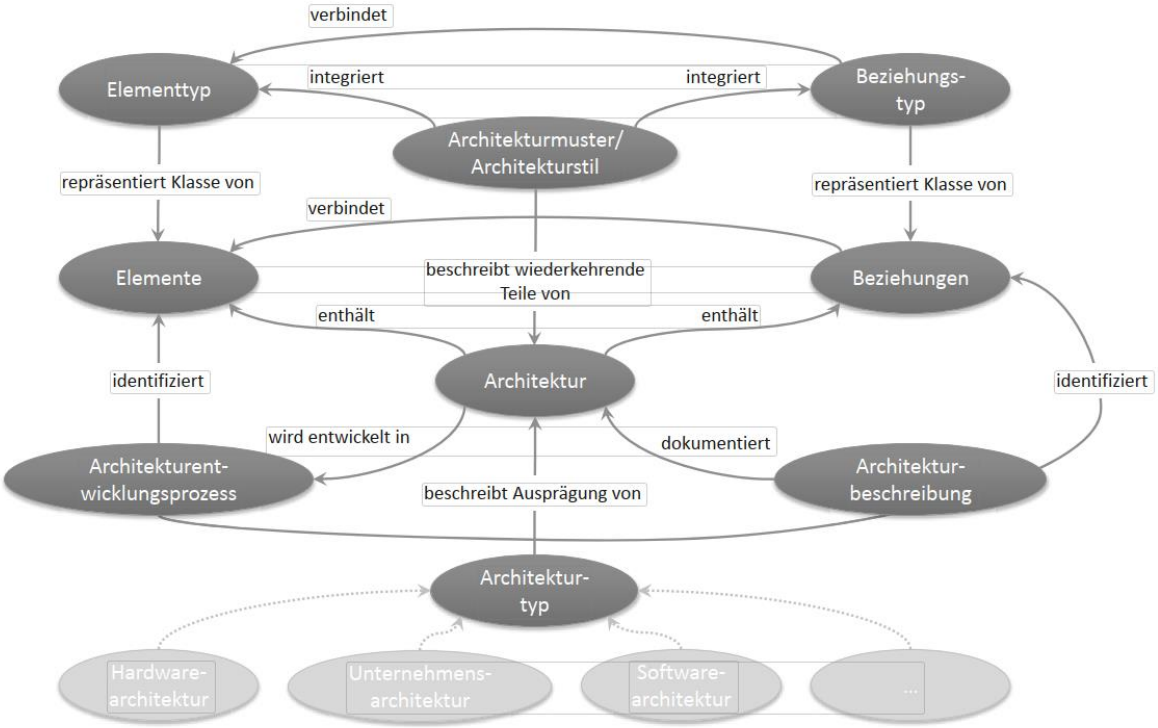


*Abbildung 1: Begrifflichkeiten im Kontext Architektur*

IEEE betont, dass eine (Software-)Architektur eine oder mehrere Strukturen umfasst, deren von außen sichtbaren Eigenschaften (z. B. Verhalten) und die Beziehungen unter diesen. Der Begriff Softwarearchitektur wird danach wie folgt definiert:

*„The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them."*

Die extern sichtbaren relevanten Eigenschaften werden als Annahmen beschrieben, die Elemente über andere Elemente treffen können, wie z. B. angebotene Dienste, Fehlerbehandlungen oder gemeinsame Ressourcennutzung, etwa von Daten. Insofern kann auch von der Schnittstellendefinition innerhalb einer Architektur gesprochen werden. Die Architektur definiert Softwareelemente (z. B. Komponenten, Module), d. h. insbesondere, dass Elemente die nicht zur Architektur bzw. Interaktion mit anderen Elementen beitragen, bewusst nicht definiert werden, da Modellbildung immer auch eine Verkürzung der Realwelt auf relevante Teile bedeuten muss.

Schließlich werden bei der Architekturentwicklung sowie der daraus resultierenden Beschreibung der Architektur wichtige (Kern-)Entwurfsentscheidungen festgehalten und Wiederverwendung ermöglicht. Durch die Architektur werden Implementierungseinschränkungen festgelegt, mit welchen die Menge der möglichen Lösungen reduziert wird. Dennoch sollte es nicht das Ziel sein eine „ideale" Architektur zu erreichen, sondern vielmehr sollte eine „gute" Architektur entwickelt werden, mit der ein System, das gemäß der Architektur entwickelt wird, die gestellten Anforderungen mit zur Verfügung stehenden Ressourcen erfüllt. Dabei sollte sie eindeutig, allumfassend, konsistent und verständlich sein.

Jedes System (Software, Hardware, etc.) besitzt IMMER eine inhärente Architektur, die dessen Strukturen und Beziehungen unterschiedlicher Typen zueinander beschreibt und sich, wenn nicht bewusst, zumindest emergent ergibt. Daher sollte bei der Architekturentwicklung eine

Vergegenwärtigung der resultierenden Architekturtypen des jeweiligen Systems stattfinden, d. h. ob es sich z. B. um Hardware, Software, einen Computer oder ein Unternehmen handelt.

Sicherlich wird eine so klare Trennung in der realen Praxis nicht immer möglich sein und entsprechende Vorgehensweisen für den jeweiligen Bereich angewandt werden. Bei der Konstruktion von Systemen wird die Architektur im Architekturentwicklungsprozess definiert und letztlich in Form einer Architekturbeschreibung expliziert, welche wiederum Vorschriften für die Realisierung des Systems darstellt. Mit einer Architekturbeschreibung wird somit das Wissen über ein System möglichst problemadäquat, d. h. in der Regel abstrakt und angemessen für die betreffenden Stakeholder, ausgedrückt. Die detaillierten, beschriebenen Strukturen respektive ein (Teil-)Modell einer Architektur kann letztlich auch die Grundlage für modellbasierte Entwicklung darstellen und die Komponentenentwicklung vereinfachen. Die wesentlichen dargestellten Begriffe und deren Zusammenhänge sind zur Veranschaulichung in Abbildung 1 dargestellt. Bei der Beschreibung und Erstellung der Strukturen und Beziehungen von Architekturen hat sich die Verwendung bestimmter, wiederkehrender (Architektur-)Muster als Konstruktions- und Kommunikationsgrundlage als vorteilhaft herausgestellt, auf die im Folgenden zur Abgrenzung kurz eingegangen werden soll

## 1.3 Wiederverwendung – das Pattern oder auch Muster

Architekturmuster (engl. architecture pattern) definieren Elementtypen und deren Interaktionen miteinander. Ein Architekturmuster ist nach ISO 42010 eine Beschreibung von Element- und Beziehungstypen zusammen mit Randbedingungen zu deren Verwendung ist. Ein Muster kann danach als eine Menge von Einschränkungen auf einer Architektur verstanden werden, die für die Elementtypen wie auch auf deren Interaktionsmustern gelten. Weiter heißt es, dass diese Einschränkungen eine Menge oder eine Familie von Architekturen bestimmen, welche diesen genügen.

Auch Architekturmuster dienen der Reduktion von Komplexität wie die Modellierung an sich, indem wiederkehrende Sachverhalte in Form eines definierten Konzeptes verwendet werden können. Sie beschreiben keine starre Lösung, sondern legen geeignete Einschränkungen für die Architektur fest und sind keine konkrete Architektur. Dennoch vermitteln sie ein nützliches Bild der Architektur und des damit verbundenen Systems. Architekturmuster werden häufig auch mit dem Begriff Architekturstil (engl. architectural style) bezeichnet – damit wird aber meist auf eine gröbere Granularität im Vergleich zu Architekturmuster abgezielt.

Beispiele für solche Architekturmuster sind beispielsweise einfache Muster wie Client-Server oder Pipes und Filter. Der letztgenannte Stil besitzt die zwei festgelegten Elementtypen Pipes und Filter. Mit diesem Architekturmuster wird eine strukturelle Einschränkung festgelegt: Pipes können mit einem Filter verbunden sein, aber Pipes können nicht mit anderen Pipes verbunden sein, wie auch Filter nicht mit anderen Filtern verbunden sein können. Weiterhin wird die Funktionalität der Architekturelemente festgelegt, insofern als dass die Verarbeitung in den Filtern stattfinden soll und die Pipes für die Datenverbindungen zuständig sind.

Beim Client-Server Muster werden ebenfalls zwei Elementtypen definiert – Client und Server. Deren Koordination wird im Sinne von Protokollen festgelegt, die der Server benutzt um mit seinen Clients zu kommunizieren, wobei ein Server i. d. R. mehrere Clients bedient. Über die Funktionalität von Client und Server oder deren Zuordnung wird keine Aussage getroffen, somit ist sie spezifisch für das entsprechende System und sollte sich in den Elementen der Architektur widerspiegeln. Nach dieser Definition sind viele Architekturen Client-Server Architekturen, die aber alle unterschiedlich sind bzw. sein können.

Durch die Architektur werden für das spätere System auch Qualitätsmerkmale wie z. B. Performanz, Sicherheit oder Wartbarkeit mitbestimmt. Architekturmuster als bewährte Strukturen besitzen bekannte Qualitätsmerkmale, was beispielsweise Vorteile für die Architekturentwicklung mit sich bringen kann. Einige Muster eignen sich besser für die zugrunde liegenden Anforderungen als andere Muster und somit kann bei der Architekturentwicklung auf bekannte Stile bspw. für hohe Performanz, hochsichere oder hochverfügbare Systeme zurückgegriffen werden. Allerdings können Qualitätseigenschaften der Software nicht allein anhand einzelner Artefakte bestimmt werden, sondern sie müssen vielmehr im Kontext der Gesamtarchitektur unter Berücksichtigung vielfältiger Beziehungen und Interaktionen betrachtet werden. Ferner ermöglicht eine Strukturierung der Architektur unter Verwendung bekannter Muster ein leichteres Verständnis für die Stakeholder der Architektur, da diese mit den Konzepten bereits vertraut sind.

## 1.4 Der Referenzbegriff

Nach der Vorstellung verschiedener Sichtweisen des Architekturbegriffes, wird im Folgenden dargestellt, wie der Referenzbegriff eingeordnet werden kann.

Als Referenz gilt im Allgemeinen ein Bezugssystem. Neben der Verwendung als Bezug oder Verweis kann der Begriff Referenz auch im Sinne einer Empfehlung verwendet werden. Die Art des Empfehlungscharakters kann jedoch unterschiedlich ausgeprägt sein. Zum einen kann etwas als Referenz deklariert werden und zum anderen kann es von den Nutzern als Referenz angesehen bzw. verwendet werden. Für die praktische Anwendbarkeit bedeutet dies, dass als Referenz deklarierte Architekturen beispielsweise höchst subjektiv sein können und von den Nutzern nicht zwangsläufig als Referenz anerkannt werden. Im anderen Fall ist der Empfehlungscharakter stärker fundiert, da sie intersubjektiv akzeptiert sind und i. d. R. auf Erfahrungen mehrerer Stakeholder beruhen.

Insbesondere kann man davon ausgehen, dass es eine Architektur im Falle nutzerseitiger Akzeptanz erst zu einer Referenz werden kann, sobald es mindestens eine Anwendung gegeben hat.

Referenzmodelle sind im näheren Sinne auch Bestandteile von Architekturen für Softwaresysteme. Eine Abgrenzung und das Verständnis im Bereich der allgemeinen Wirtschaftsinformatik sowie der Softwarearchitektur sollen im Folgenden kurz dargestellt werden.

In der Wirtschaftsinformatik wird der Referenzmodellbegriff im Rahmen der Informationssystementwicklung geprägt. Nach Becker ist ein Referenzmodell (bzw. ausführlich Referenzinformationsmodell) ein Informationsmodell, das zur Unterstützung der Konstruktion von anderen Modellen genutzt wird. Damit dies möglich ist, muss es von individuellen Besonderheiten abstrahieren und allgemeingültig sein. Die Allgemeingültigkeit bezieht sich hierbei nicht auf eine universelle Gültigkeit, sondern nur auf die Gültigkeit in Bezug auf eine Klasse von Anwendungsfällen. Dabei wird keine Aussage darüber gemacht, ob die Klasse von Anwendungsfällen innerhalb einer Domäne verortet ist (bzw. eine Domäne darstellen) oder ob diese übergreifend ist.

Überträgt man diese Definition auf einen Referenzarchitekturbegriff, sollte eine Referenzarchitektur sinngemäß als eine Referenz zur Konstruktion einer Architektur für eine konkrete Lösung herangezogen werden können. Der spezielle Unterschied von Referenzarchitekturen zu anderen Architekturen liegt dabei darin, dass sie von Besonderheiten abstrahieren und über (Allgemein-) Gültigkeit für eine Klasse von Anwendungsfällen verfügen. Die Art der Empfehlung, die mit einer Referenzarchitektur einhergeht – ob nutzerseitig oder dazu benannt – erlaubt gegebenenfalls auch Rückschlüsse auf die praktische Relevanz und Anwendbarkeit der Architektur.

Im Rahmen der Softwareentwicklung definiert sich der Begriff einer Referenzarchitektur (genauer Referenzsoftwarearchitektur) auf Basis der Definition der IEEE zu Softwarearchitekturen:

*„Eine Referenzarchitektur ist eine abstrakte Software-Architektur, sie definiert Strukturen und Typen von Software-Elementen sowie deren erlaubte Interaktionen und ihre Verantwortlichkeiten speziell für einen Anwendungsbereich. Die Strukturen sind jeweils für alle Systeme innerhalb einer Domäne anwendbar.“*

Als Charakteristikum kann hier der Aspekt der Abstraktheit der Referenz gesehen werden, was auch durch die Einschränkung auf Typen von Software-Elementen unterstützt wird. Zudem wird die Allgemeingültigkeit für eine bestimmte, eng umgrenzte Domäne definiert, die eine Referenzarchitektur besitzt, was bedeutet, dass diese innerhalb der Domäne auf mehrere Systeme anwendbar ist. Eine Abgrenzung zur Definition der Architekturmuster besteht hier im Umfang einer Referenzarchitektur, insofern als dass Architekturen mehrere Strukturen (wie z. B. Architekturmuster) enthalten können und dass sie stets domänenspezifisch etwa im Kontext GAIA-X Federation Services und Data Spaces sind.

Eine Referenzarchitektur ein Referenzmodell, das auf Softwareelemente abgebildet wird, die kollaborativ die Funktionalität des Referenzmodells sowie die Datenflüsse zwischen den Elementen realisieren. Ein Referenzmodell unterteilt die Funktionalität und die notwendigen Datenflüsse zwischen den Teilen, was mit einer Referenzarchitektur auf eine Systemdekomposition abgebildet wird. Die Aufteilung der Funktionen auf Softwareelemente kann dabei auf verschiedene Arten stattfinden, d. h. einzelne Softwareelemente können Teile einer Funktion oder aber auch mehrere Funktionen realisieren. Weiterhin wird gesagt, dass Referenzmodelle Standard-Dekompositionen eines bekannten Problems in Teile bieten, die gemeinsam das Problem lösen. Als Beispiele dafür werden Referenzmodelle für Compiler oder Datenbanksysteme angeführt, deren Standard-Komponenten und das Zusammenspiel darin beschrieben werden. Referenzmodelle entstehen den Autoren zufolge aus Erfahrung und stellen ein Charakteristikum für weit entwickelte Anwendungsbereiche dar. Insofern findet sich hier eine klare Aussage bezüglich des Empfehlungscharakters und der Gültigkeit in einer festgelegten Domäne wieder.

Eine pragmatischere Definition jedoch zumeist gewählt. Der Wortteil Referenz wird dort im Sinne eines Referenz-Buches (wie z. B. Wörterbücher, Enzyklopädien oder Kompendien, aber auch Service Repositorien) eingeführt, welche den Nutzern eine gut durchsuchbare, organisierte Struktur bieten, um bestimmte Fragestellungen mittels vereinten Expertenwissens zu beantworten. Darin sind viele Querverweise enthalten, so dass ein Benutzer zwischen vielen verschiedenen Einträgen navigieren kann.

Der Architekturbegriff wird hier wie folgt verwendet. Eine Referenzarchitektur für eine unternehmensweite Infrastruktur ist anhand häufiger Typen von Anforderungen, die über viele verschiedene Funktionsbereiche und Systeme hinweg gefunden werden können, sowie nach sogenannten „high-level“ Konzepten der Systementwicklung organisiert.

Zusammenfassend sollen nun einige Merkmale und Gemeinsamkeiten dieser Definitionen in einem zusammengesetzten Referenzarchitekturbegriff vereint werden:

Eine Referenzarchitektur beschreibt den Aufbau eines Systems mit seinen Elementtypen und -strukturen sowie deren Interaktionen untereinander und der Umwelt. Damit werden Einschränkungen bezüglich der Instanziierung der Architektur festgelegt. Individuelle Besonderheiten werden dabei abstrahiert und somit zeichnet sich eine Referenzarchitektur durch

- Allgemeingültigkeit in einer speziellen Domäne aus. Mit einer Referenzarchitektur kann so
- die Konstruktion weiterer Architekturen mit dem funktionalen Anspruch der Referenzarchitektur unterstützt werden. Dies kann auch die Definition von Leitlinien für die Verwendung, Evolution und Verantwortlichkeiten umfassen.
- Darüber hinaus besitzt eine Referenzarchitektur einen Empfehlungscharakter, der sich aufgrund von Erfahrungen sowie einer breiten Nutzerakzeptanz und -anerkennung herausbildet oder festgesetzt wurde.

Ein Verständnis von solchen, aus Erfahrung gewachsenen Referenzarchitekturenist ein Kern der Wiortschaftsinformatik. Referenzarchitekturen entstehen durch die Beschreibung von „besten Lösungen“, die nach Beurteilung von Gemeinsamkeiten verschiedener Architekturen durch Experten unter Berücksichtigung der Anforderungen in Übereinstimmung mit „high-level“ Konzepten bestimmt werden. Letztlich unterscheidbar von gewöhnlichen Softwarearchitekturen ist eine Referenzarchitektur durch das Merkmal der Abstraktheit.

Ferner wird durch die IEEE betont, dass Referenzmodelle, Architekturmuster und Referenzarchitekturen keine Architekturen sind, sondern nützliche Konzepte, die Elemente einer Architektur festhalten. Aus Referenzarchitekturen entstehen, ähnlich wie bei Architekturmustern, verschiedene Architekturen die der strukturellen Vorgabe der Referenzarchitektur genügen. Referenzarchitekturen unterscheiden sich aber von Architekturmustern durch eine detailliertere Ausarbeitung sowie eine spezifische Anwendungsdomäne bzw. Betrachtungsgegegenstand. Insofern kann das Entwickeln einer Referenzarchitektur im Vorfeld einer Architekturentwicklung auch zur gezielten Prägung späterer Architekturen in der Domäne im Kontext eines emerging Topics auf niedrigem TRL dienen. Die Arbeiten der IDSA im Kontext IDS-RAM sind hierbei zu nennen.

Eine Architektur gilt als konform zu einer Referenzarchitektur, wenn sich die entsprechenden Elementtypen und -strukturen sowie deren Interaktionen der Referenzarchitektur darin wiederfinden. Das bedeutet, dass die Architektur Elemente definiert, die entsprechend der Vorgaben der Referenzarchitektur in Beziehung stehen. Da eine Architektur verschiedenste Strukturen definieren kann, kann sie auch zu mehreren Referenzarchitekturen konform sein (vgl. hier auch ISO 42010 als Architekturmetamodell):

Eine Einteilung kann dazu dienen, einen Überblick über mögliche Arten von Referenzarchitekturen zu erhalten, die zur weiteren Einordnung und Abgrenzung auf oberstem Niveau herangezogen werden können.

## 1.5 Dokumentationserfordernisse

Nachdem an den Begriff der Referenzarchitektur herangeführt wurde, wird nun die Dokumentation einer Referenzarchitektur näher betrachtet. Die Beschreibung der Architektur stellt das wesentliche Artefakt dar, welches als Grundlage für die an der Entwicklung und Realisierung des Systems Beteiligten sichtbar ist. Die Dokumentation hilft dabei die Entscheidungen für die Aufteilung des Problembereiches nachzuvollziehen und bildet zudem z. B. die Grundlage für benötigte Schnittstellen, Konformitätstests sowie das Verständnis und Management abgeleiteter Architekturen. Durch die Referenzarchitektur bzw. deren Dokumentation werden bindende Einschränkungen für die daraus resultierenden Entwicklungen festgelegt. Daher ist es notwendig, dass die Dokumentation dies möglichst optimal nutzbringend unterstützt, und die getroffenen Entscheidungen unter der Einhaltung bestimmter Qualitätskriterien, wie beispielsweise Exaktheit, Konsistenz und Vollständigkeit festhält.

Um Fehlinterpretationen und Missverständnisse zu vermeiden, ist für die Dokumentation eine möglichst eindeutige Sprache notwendig. Informelle und unklare Aussagen in der Dokumentation hingegen könnten eine falsche Sicherheit erwecken und damit zu fehlgeleiteten Entwicklungen führen.

Darüber hinaus muss die Dokumentation den Anforderungen der Verwender/ Stakeholder Rechnung tragen, damit diese auch die benötigten Informationen entnehmen können.

Referenzarchitekturen werden von verschiedenen Beteiligten zu unterschiedlichen Zwecken verwendet. So können sie von Architekten herangezogen werden, um beispielsweise ein zu entwickelndes System grundlegend zu strukturieren und um die dort definierten Begriffe zur Kommunikation der entsprechenden Architektur z. B.mit Entwicklern zu verwenden. Für das Projektmanagement kann eine Referenzarchitektur zur Planung benötigter Qualifikationen von Projektmitarbeitern und zur initialen Aufwandsschätzung dienen. Zudem kann eine Referenzarchitektur auch Lösungsherstellern dienen, um konforme Systeme zu entwickeln. Bedingt durch die unterschiedlichen Stakeholder, sollte eine Referenzarchitektur über entsprechende Inhalte verfügen, um zweckmäßig zu sein. Diese Inhalte sind daher nicht starr vorgegeben, sondern letztlich abhängig von der Zielgruppe, welche die Referenzarchitektur verwendet. Eine Empfehlung an die minimalen Inhalte von Architekturbeschreibungen findet sich in der ISO 42010 wo weiterhin grundlegende Begriffe im Kontext der Dokumentation softwareintensiver Systeme beschrieben werden (insbesondere das „conceptual framework for architecture description“). Diese Inhalte umfassen u. a. eine eindeutige Kennzeichnung und Übersicht des Dokumentes, Identifikation von Stakeholdern und deren architekturrelevante Interessen, Spezifikationen von ausgewählten Standpunkten, von denen aus die Architektur betrachtet werden soll, entsprechende Architekturdarstellungen, Aufzeichnungen bekannter Inkonsistenzen sowie die Dokumentation von Entwurfsentscheidungen.

Zur Beschreibung einer Referenzarchitektur können im Wesentlichen die Mittel der herkömmlichen Architekturbeschreibung herangezogen werden, unter Berücksichtigung der allgemeingültigen Strukturen und Beziehungen. Dabei ist es ein Leitsatz, dass Architekturen am besten durch mehrere Sichten (Views) ausgedrückt werden. Zur Beschreibung Software-intensiver Systeme existieren verschiedene Ansätze, wie z. B. der IEEE-Standard 1471-2000 [41] oder der „Views and Beyond“-Ansatz des Software Engineering Institutes die einen View-orientierten Ansatz verfolgen. Im Weiteren soll der IEEE Standard 1471-2000 betrachtet werden, der sich derzeit bei der ISO und IEEE als ISO/IEC 42010 in der Weiterentwicklung und Normierung befindet. Der darin definierte Begriffsapparat (dargestellt in Abbildung 2) und vorgeschlagene Inhalte zur Architekturbeschreibung sollen im Folgenden kurz eingeführt werden.

Eine Architekturbeschreibung besteht danach aus einer Sammlung verschiedener Produkte, die eine Architektur dokumentieren. Welche Artefakte zur Dokumentation herangezogen werden, hängt dabei von den Stakeholdern des Systems (Stakeholder) ab. Dies können z. B. Auftraggeber, Benutzer, Entwickler oder Maintainer des Systems sein, welche jeweils verschiedene oder komplementäre Interessen (Concerns) haben können. Das vorrangige Interesse eines Benutzers wird beispielsweise Informationen bezüglich der Benutzung des Systems gelten, wohingegen Entwickler eher an technischen- und Implementierungs- Details interessiert sein werden.

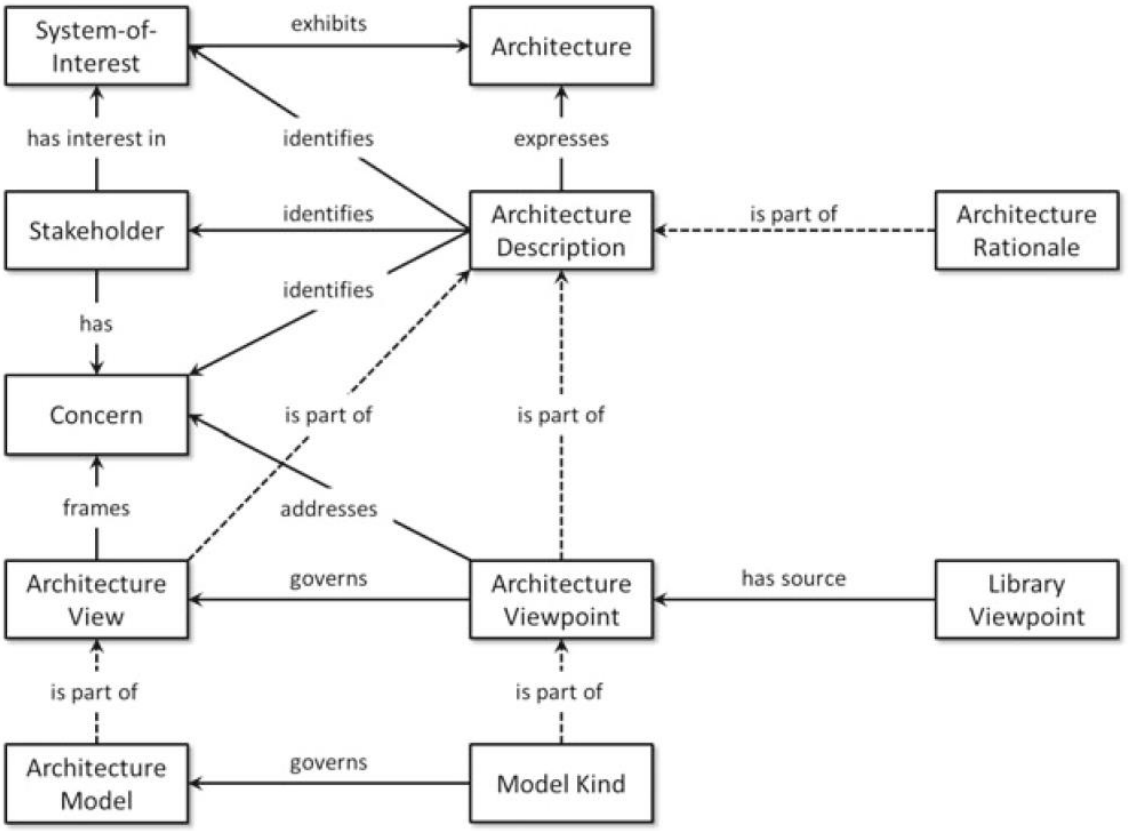


*Abbildung 2: ISO IEEE/IEC 42010 Metamodell*

Die Architektur (Architecture) eines Systems (System-of-Interest) wird in einer Architekturbeschreibung (Architectural Description) mittels mehrerer Sichten (Architecture View), in diesem Sinne multiperspektivistisch, dokumentiert.

Eine einzelne Sicht stellt dabei das Gesamtsystem bezogen auf die spezifischen Inhalte dar. Eine Gesamtarchitektursicht ergibt sich somit aus den subjektiv relevanten Sichten (der einzelnen Stakeholder) auf die Architektur. Ein „Architecture View“ kann aus einem oder mehreren Architekturmodellen (Architecture Model) bestehen. Diese Modelle können auch Teil mehrerer Views sein. Damit die unterschiedlichen Interessen der Stakeholder abgedeckt werden, gilt es zunächst diese unterschiedlichen Sichtweisen (Architecture Viewpoint) zu bestimmen und zu dokumentieren. Bei der Definition von Architecture Viewpoints wird festgehalten, für welchen Stakeholder diese interessant sind, deren typische Anliegen (Concern) und wie diese Sichten aufgebaut/modelliert sein sollen (Model Kind). Darunter fallen z. B. auch welche Inhalte, sowie Modellierungs- und Analysetechniken ausgewählt werden sollen. Eine detailliertere Beschreibung der minimal an die Beschreibung geforderten Inhalte kann in der Norm nachgelesen werden. Ein Viewpoint stellt in diesem Sinne eine Vorlage dar, anhand derer Views erstellt werden sollen. Ein Viewpoint „Grundriss“ aus der Gebäudearchitektur soll dazu dienen dies näher zu illustrieren.

Mit dem Grundriss sollen Grundfläche, Positionen der Wände im Gebäude sowie Aussparungen für z. B. Versorgungsschächte oder Treppenhaus in einer Aufsicht für ein Stockwerk maßstabsgetreu dargestellt werden. Stakeholder, die an diesem Viewpoint interessiert sein können, sind z. B. Bauträger und ausführende Arbeitskräfte aus Gründen der Raumplanung und -erstellung. Ein dazu konformer View ist dann eine Zeichnung für ein konkretes Gebäude, wie er wohl bei jedem Neubau angefertigt

werden dürfte. Verständlich wird hier auch, dass ein einziger Viewpoint nicht ausreicht, um eine ganze Architektur zu dokumentieren (hier erfolgt beispielsweise keine Darstellung der Raumhöhen oder Versorgungsleitungen). Verwendet man jedoch mehrere Viewpoints um eine Architektur darzustellen, ist es unbedingt erforderlich, dass den sich daraus ergebenden Sichten ein übergreifendes, einheitliches Modell zugrunde liegt. Die Modellierungsform und Darstellung muss dazu zudem so gewählt sein, dass sie von den entsprechenden Stakeholdern eindeutig verstanden wird, damit diese aus ihrer Perzeption der Architektursicht entsprechende Handlungen aneinander ausrichten können, was insgesamt für eine formale Darstellungsform spricht.

Als Darstellungsmöglichkeiten für Softwarearchitekturen werden im einfachsten Fall häufig informelle so genannten „Boxen- und Liniendiagramme“ genannt, welche auch die Unified Modeling Language (UML) mit einigen ihrer Diagrammtypen im erweiterten Sinne zur Verfügung stellt.

Der IEEE-Standard macht keine Vorgaben bezüglich zu verwendender Viewpoints oder deren Inhalten. Existierende, aufeinander abgestimmte, dokumentierte und wiederzuverwendende Viewpoints werden als sogenannte Library Viewpoints bezeichnet. Diese können über mehrere Beschreibungen hinweg, i. d. R. mit kleineren Anpassungen, wiederverwendet werden.

Das Referenz-Modell für „Open Distributed Processing“ (RM-ODP) definiert beispielsweise fünf solcher Viewpoints, die für die Betrachtung offener, verteilter Systeme von Bedeutung sind (Enterprise Viewpoint, Information Viewpoint, Computational Viewpoint, Engineering Viewpoint und Technology Viewpoint). Darüber hinaus sind Viewpoints Bestandteil von Architektur-Frameworks, wie z. B. TOGAF.

# 2 RAMI – Referenzmodell Industrie 4.0

Im Kontext des Themenfeldes Architekturstandards in I4.0 sind vor allem Aspekte der Wiederverwendung des Wissens einzelner technischer Umsetzungen der Anwendungsfälle relevant ([1]). (IT-)Architekturstandard ist ein Sammelbegriff für alle Aspekte der IT in einer Organisation – ob nun statisch in Form von Hardware oder dynamisch in Form von Software. Hierfür haben sich in anderen Domänen so genannte Referenzarchitekturen als Dokumentationsstandard als Blaupausen etabliert.

Im Besonderen im Kontext der Integration von OT (Operational Technology) mit IT (Information Technology) ist die Interoperabilität der Datenprozesse im Fokus dieser Architekturen. Im Smart Grid hat sich beispielsweise das so genannte SGAM – Smart Grid Architecture Model etabliert [3]. Das SGAM und seine Methodik sollten ursprünglich dazu dienen, das Design und die Technologien von Smart Grid-Anwendungsfällen sowohl architektur- als auch technologieneutral zu (re-)präsentieren – eine Anforderung, die es auch in InGAIA-X gibt.

Darüber hinaus ermöglicht die dreidimensionale Sichtweise eine Trennung der verschiedenen Sichtweisen in der Systementwicklung. Das SGAM musste für praktische Zwecke entwickelt werden. Im Versorgungsbereich hatten die Anbieter ein Monopol auf den Verkauf monolithischer Systeme an ihre Kunden, die aufgrund ihrer gesicherten Einnahmen in Form von regulierten Unternehmensgewinnen gewohnt waren, hohe Preise zu zahlen. Mit der Energiewende führte ein neues Betriebsparadigma zu einer Vielzahl neuer Schnittstellen zu verteilten Systemen, Sensoren, erneuerbare Einspeiser und mehr Betriebsdaten, die aufgrund von Betriebsoptimierungserfordernissen gesammelt werden. Das Thema Smart Grid kam auf. Die Versorgungsunternehmen waren nicht bereit für das neue IT-Paradigma und die benötigten Technologien und die daraus resultierenden Anforderungen. Ihre IT-Abteilungen verfügten zumeist nur über geringe Kenntnisse in den Bereichen konzeptionelle Modellierung, Management von

Unternehmensarchitekturen, Systemtechnik und serviceorientierte Architekturen, da sie sich bei schlüsselfertigen Lösungen in der Regel auf Integratoren, OEMs (Original Equipment Manufacturer) und Anbieter verließen. Benutzerdefinierte Schnittstellen, die auf neuen Benutzeranforderungen für neue Geschäftsprozesse basieren, wurden zu einer Herausforderung.

Es existierten keine domänenspezifischen Unified Modeling Language UML-Tools, PowerPoint und Visio waren die Mittel der Wahl für die Modellierung von IT -Landschaften und es wurde kein Wert auf eine standardisierte Semantik und Syntax gelegt. Die Einführung der Modellierung für Smart Grids erforderte also das Erlernen aller domänenspezifischen Anforderungen, proprietärer Werkzeuge und Prozesse für den mit Informatik vertrauten Integrator. Der Fokus musste auf einer Methode liegen, die ihre theoretische Grundlage (z.B. der ISO / IEC 42010 Meta-Modellierung für Architekturen Standard) verbirgt und den Aspekt des organisatorischen Umfangs, der IT-Ebene und der Wertschöpfungskette für den Stakeholder beim Energieversorger nachvollziehbar macht. Generell war der Weg von einer einfachen Anwendungsfallvorlage mit einem iterativen (und ggf. agilen) Prozess zur Codegenerierung unter Berücksichtigung des begrenzten Modellierungshintergrunds der Stakeholder und ihrer Domänenanforderungen notwendig.

Das SGAM besteht aus fünf konsistenten und kanonisch verwendeten Schichten, die Anwendungsfälle, Informationsmodelle, Kommunikationsprotokolle und Komponenten darstellen. Jede Schicht deckt eine so genannte Smart Grid-Ebene ab, die von Domänen (der Wertschöpfungskette) aufgespannt wird. Die Absicht des Referenzbezeichnungsmodells ist es, die Darstellung des aktuellen Stands der Implementierungen im elektrischen Stromnetz zu ermöglichen, aber auch die Entwicklung zu zukünftigen Smart Grid-Szenarien darzustellen, indem es in seinem Kern die Prinzipien der Erweiterbarkeit, Skalierbarkeit und Aufrüstbarkeit unterstützt. Die im SGAM modellierten Bereiche berücksichtigen hauptsächlich die Energieumwandlungskette. Die Hierarchie der Verwaltung und des Betriebs von Energiesystemen aus Sicht der Automatisierung und der Organisationsebene der Versorgungsunternehmen wird im SGAM durch die Definition weiterer Bereiche widergespiegelt. Schließlich definiert das SGAM als eine wichtige Anforderung an verteilte Systeme sogenannte Interoperabilitätsschichten, die auf dem GWAC (GridWise Architecture Council) IOP (Interoperability)-Stack basieren. In diesem Zusammenhang sind im System Engineering verschiedene neue SGAM-basierte Modelle entwickelt worden. Diese Modelle konzentrieren sich auf das Home and Building Architecture Management (HBAM), das so genannte Smart City Infrastructure Architecture Model SCIAM, das Electric Mobility Architecture Model (EMAM), das Maritime Architecture Framework (MAF) und natürlich das RAMI 4.0, das in URANOS-X ausgewählt wurde, eine Begründung hierzu findet sich später im Abschnitt. Der RAMI Ansatz trägt der Notwendigkeit Rechnung, die Anforderungserhebung besser zu strukturieren [3].

Ferner ermöglicht das RAMI 4.0 Modell es auch, geeignet Operational Technology (OT) Datenquellen, welche heutzutage in Verwaltungsschalen gekapselt sein können, zu modellieren und auf der übergeordneten IT (Information Technology) Ebene diese Prozessdaten wiederum mittels GAIA-X Diensten in einem Data Space zur Verfügung zu stellen. Data Spaces können dann auf dem Business oder Function Layer modelliert werden.

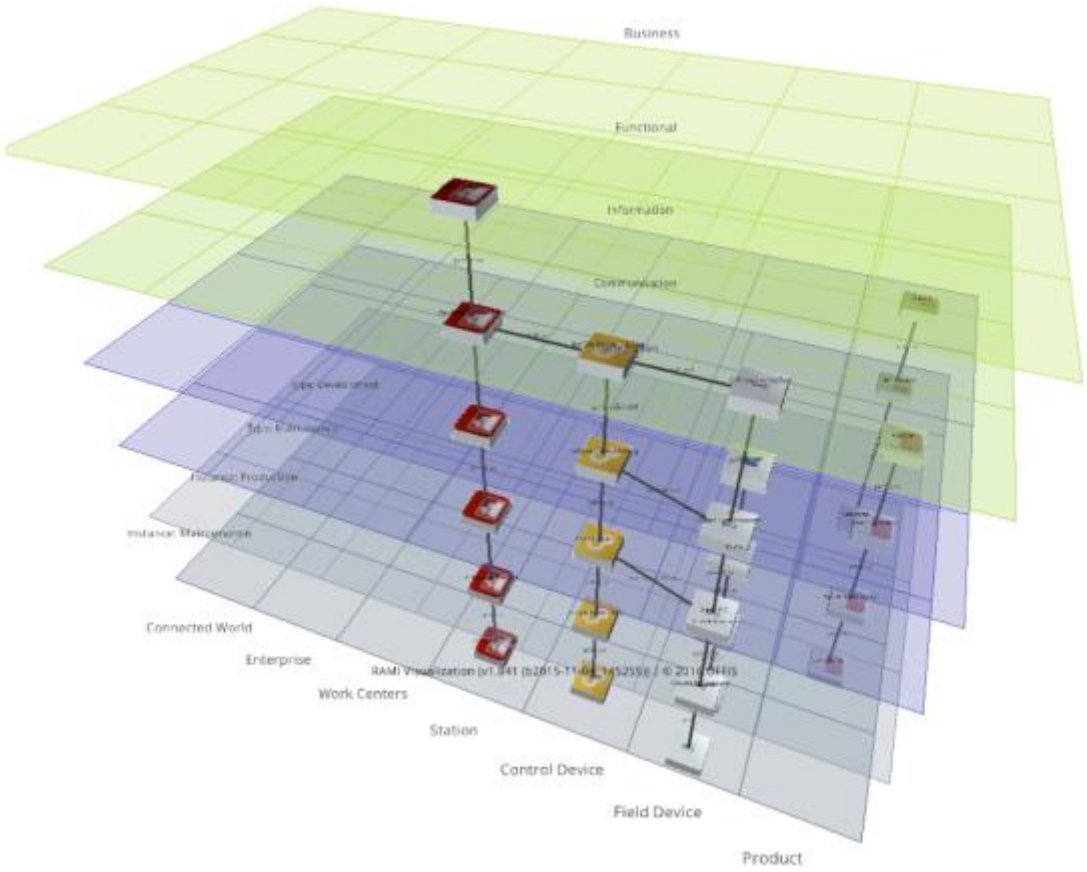


*Abbildung 3: RAMI 4.0*

Durch eine strukturierte Anforderungserhebung mit Hilfe eines IEC 62559 Templates oder auch einem Business Model Canvas kann sichergestellt werden, dass genügend Informationen erhoben werden, um ein RAMI 4.0-Architekturmodell eines (technischen) Industrie 4.0-Anwendungsfalls zu erstellen. Beide Werkzeuge nutzen das gleiche ISO/IEC 42010-Metamodell und sind aufeinander abgestimmt (vgl. Abbildung 4). Mit der gesicherten Datenqualität können die technischen Lösungen und ihre inhärente Architektur für Diskussionen mit den Beteiligten oder für graphische Analysen visualisiert werden. Schnittstellen, ausgetauschte Daten sowie Application Programming Interfaces (API) werden dokumentiert und zur Diskussion mit den Stakeholdern gestellt. Da dieser Schritt in einer frühen Phase eines Projekts stattfindet, senkt die Harmonisierung und Abstimmung der Anforderungen sowohl für funktionale als auch für nicht-funktionale Aspekte die Integrations- und Entwicklungskosten. Wichtige Informationen zu Schnittstellen und

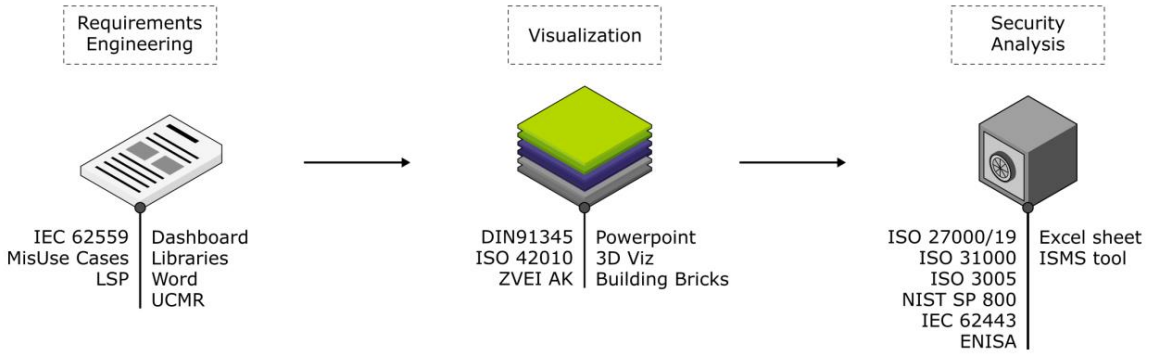


*Abbildung 4: Mögliche Toolchain für RAMi 4.0 Engineering*

Datenaustausch wie technische Standards, Nutzlasten, Quality-of-Service, Uptime etc. können diskutiert und bewertet werden.

Das RAMI 4.0 deckt andere Dimensionen ab als das SGAM, aber die ursprüngliche Methode und Toolchain kann an diese Änderungen angepasst werden.

Das RAMI 4.0-Derivat des ursprünglichen SGAM-Domänenansatzes folgt typischerweise den Interoperabilitäts-Stack-Regeln für die technischen Lösungen, wie sie von SGAM definiert wurden, weil diese aus der Stakeholder-Perspektive einzunehmenden Gesichtspunkte in der Regel eine natürliche Passung für (technische) Entwicklungsprojekte im Kontext automatisierter Systems-of-Systems sind. Einige grundlegende Designregeln für die Modellierung der geplanten technischen Anwendungsfälle in den drei Dimensionen müssen befolgt werden, damit verschiedene Metriken, Styleguides und Werkzeuge mit dem neuen Modell arbeiten können. Der vorgestellte Ansatz macht sich die Tatsache zunutze, dass das RAMI viele / die meisten grundlegenden Gestaltungsprinzipien mit dem SGAM teilt, wodurch eine Übertragung der SGAM-Visualisierung auf den Bereich Industrie 4.0 möglich wird. Dieser Abschnitt bietet eine sehr kurze Einführung in das RAMI-Modell und konzentriert sich dabei auf die wichtigsten Änderungen aus Sicht des Modellierungsparadigmas im Requirements Engineering und Domain Engineering.

Das Referenzarchitekturmodell für Industrie 4.0 (RAMI 4.0) ist das derzeit anspruchsvollste Derivat des SGAM, das vom ZVEI in Deutschland entwickelt wurde. Basierend auf der diskutierten deutschen Industrie 4.0-Konzept ist der Hauptaspekt die Wiederverwendung des GWAC-Interoperabilitäts-Stacks, der die Automatisierungspyramide darstellt. Zusätzlich zu den Komponenten, Business, Funktion, Information, Kommunikation und Asset wird eine neue Schicht namens Integration (für die Industrie 4.0-Komponenten-Meta-Informationen und die Verwaltungsschale) eingeführt. Das Modell soll die verschiedenen Nutzerperspektiven auf das Thema Industrie 4.0 auf der Gesamtebene harmonisieren und ein gemeinsames Verständnis für die Stakeholder über die Beziehungen zwischen den einzelnen Komponenten für Industrie 4.0-Lösungen von verschiedenen Anbietern schaffen. Unterschiedliche industrielle Teilbereiche wie Automatisierung, Maschinenbau und Verfahrenstechnik müssen sich auf eine gemeinsame Sicht auf die Gesamtsystemlandschaft einigen. Die SGAM-Prinzipien für den Umfang der Lokalisierung von Standards werden in den RAMI-Paradigmen wiederverwendet, wobei auch RAMI 4.0 als Referenzbezeichnungssystem (siehe Abbildung 3) verwendet wird.

## 2.1 Empfehlungen

Im Kontext von URANOS-X hat sich in der Analyse das RAMI als geeignetes Modell zur Modellierung der Anwendungsfälle erwiesen. Kombiniert mit SysML (System Modeling Language) 2.1 bietet es passende Instrumente, Wissen verständlich aufzubereiten, vermitteln, aber vor allem auch als Referenz im Kontext Leader, Learner, Listener zu bewahren. Ferner ist der Bezug des RAMI im Kontext

industrieller Fertigung und den dazugehörigen Daten gegeben, anders am beim SGAM ist das RAMI dabei sowohl geeignet, Blueprint Technologien als auch Instanzen in einer Reference Designation System abzubilden. Dies ermöglicht es, sowohl den Entwurf als auch das finale Produkt mit seinen Abweichungen zu modellieren und ggf Varianten von Datenquellen etc. einzupflegen.

# 3 Literatur

[1] Bundesministerium für Wirtschaft und Klimaschutz (BMWK). Das Projekt GAIA-X – Eine vernetzte Dateninfrastruktur als Wiege eines vitalen, europäischen Ökosystems. Bundesministerium für Wirtschaft und Klimaschutz, Berlin, 2019.

[2] Hans-Jürgen Appelrath, Petra Beenken, Ludger Bischofs, and Mathias Uslar. IT-Architekturentwicklung im Smart Grid: Perspektiven für eine sichere markt-und standardbasierte Integration erneuerbarer Energien, 2012.

[3] Mathias Uslar, Andre Göring, Roland Heidel, Christian Neureiter, Dominik Engel, and Stefan Schulte. An open source 3d visualization for the rami 4.0 reference model. In VDE Konferenz 2016, pages 1–6, 01 2016